

## **FFI RAPPORT**

### **INTERACTING MULTIPLE MODEL PROBABILISTIC DATA ASSOCIATION – En studie av målfølgingsalgoritme for aktiv sonar**

GJERSØE, Amund Lorentz

**FFI/RAPPORT-2003/01001**



FFIBM/849/116

Godkjent  
Horten 17. mars 2003

Elling Tveit  
Forskningsjef

**INTERACTING MULTIPLE MODEL  
PROBABILISTIC DATA ASSOCIATION –  
En studie av målfølgingsalgoritme for aktiv sonar**

GJERSØE, Amund Lorentz

FFI/RAPPORT-2003/01001

**FORSVARETS FORSKNINGSINSTITUTT**  
**Norwegian Defence Research Establishment**  
Postboks 25, 2027 Kjeller, Norge



**FORSVARETS FORSKNINGSPENNINGSTUTT (FFI)**  
**Norwegian Defence Research Establishment**  
 P O BOX 25  
 NO-2027 KJELLER, NORWAY

**UNCLASSIFIED**

**SECURITY CLASSIFICATION OF THIS PAGE**  
 (when data entered)

**REPORT DOCUMENTATION PAGE**

1) <b>PUBL/REPORT NUMBER</b> FFI/RAPPORT-2003/01001 1a) <b>PROJECT REFERENCE</b> FFIBM/849/116	2) <b>SECURITY CLASSIFICATION</b> UNCLASSIFIED 2a) <b>DECLASSIFICATION/DOWNGRADING SCHEDULE</b> -	3) <b>NUMBER OF PAGES</b>  41
4) <b>TITLE</b> INTERACTING MULTIPLE MODEL PROBABILISTIC DATA ASSOCIATION – En studie av målfølgingsalgoritme for aktiv sonar  INTERACTING MULTIPLE MODEL PROBABILISTIC DATA ASSOCIATION – A study of a target tracking algorithm for use in an active sonar system		
5) <b>NAMES OF AUTHOR(S) IN FULL (surname first)</b> GJERSØE, Amund Lorentz		
6) <b>DISTRIBUTION STATEMENT</b> Approved for public release. Distribution unlimited. (Offentlig tilgjengelig)		
7) <b>INDEXING TERMS IN ENGLISH:</b> a) <u>IMM</u> b) <u>PDA</u> c) <u>active sonar</u> d) <u>target tracking</u> e) <u>clutter</u>		
<b>THESAURUS REFERENCE:</b>		
8) <b>ABSTRACT</b> This document conducts a study of a target tracking algorithm, based on interacting multiple models in combination with probabilistic data association, derived, presented and discussed in the UNISON 2005 project. The effects on the validation region and the track properties, by adjusting the parameter settings, are studied. A proximity filter is derived, presented and discussed. Questioned is the use of inverted range as a state. This is derived from systems using passive sonars, but the algorithm is for a system based upon using active sonars.		
9) <b>DATE</b>  17. March 2003	<b>AUTHORIZED BY</b> This page only  Elling Tveit	<b>POSITION</b>  Director of Research

ISBN 82-464-0719-8

**UNCLASSIFIED**

**SECURITY CLASSIFICATION OF THIS PAGE**  
 (when data entered)



**INNHold**

	<b>Side</b>	
1	INNLEDNING	7
2	BAKGRUNNSTEORI	9
2.1	Målfølgeralgoritme	9
2.1.1	Enkel forklaring av målfølgeralgoritmen	9
2.1.2	Kalmanfilter	10
2.1.3	Vekting av ekko	11
2.1.4	Vekting av track	12
2.2	Proximity Filter	13
2.3	Akronymer	13
3	ARBEID MED MÅLFØLGINGSALGORITME	15
3.1	Parameterinnstillinger	15
3.1.1	Grafer og tall som gir grunnlag for analyse	16
3.1.2	Innvirkning av Target Detection Probability	21
3.1.3	Valideringsregion og innvirkning av <i>Gate Probability</i>	24
3.1.4	Innvirkning av tillatt hastighet	27
3.1.5	Innvirkningen av målestøy	27
3.1.6	Innvirkning av Markov chain probability	29
3.2	Forslag til videre arbeid på målfølgingsalgoritmen	30
3.2.1	Tilstandsvariable	30
3.2.2	Parametere	30
3.2.3	Algoritmen	30
4	PROXIMITY FILTER	33
4.1	Frittstående proximity filter	33
4.2	Integrert proximity filter	34
5	KONKLUSJON	37
	LITTERATUR	38
	APPENDIKS A	39
o	Endringer på original målfølgingsalgoritme	39
	APPENDIKS B	40
o	Lagring av målfølgerdata for hvert ping	40
	FORDELINGSLISTE	41





## **INTERACTING MULTIPLE MODEL PROBABILISTIC DATA ASSOCIATION – En studie av målfølgingsalgoritme for aktiv sonar**

### **1 INNLEDNING**

Rapporten beskriver studiet av målfølgingsalgoritmen IMMPPDA, som er utarbeidet i (7) og videreutviklet i (5). I tillegg beskrives arbeidet med utvikling av et proximity filter for måldata, utført av undertegnede for prosjekt SIMSON. SIMSON er et prosjekt som har som mål å utvikle et analyse og simuleringsverktøy for Fridtjof Nansen-klassens Anti-Ubåt-sensorer. Det er i denne sammenhengen nødvendig å få kjennskap til målfølgingsalgoritmer for systemer med aktiv sonar.

I prosjektet UNISON 2005 ble det utviklet målfølgingsalgoritmer for å følge undervannsmål ved hjelp av sonar. Denne rapporten tar for seg et studie av én av algoritmene, IMMPPDA.

Kildekoden til algoritmen vil bli undersøkt for å sikre at teori og praksis sammenfaller, slik vil undertegnede få god innføring i både teori og implementasjon. Så vil parametrene bli testet og optimalisert. De data som er brukt i arbeidet er opptak fra Bjørnafjorden i april 1999, og er de samme som de det ble referert til i Anders Lølands (5) rapport.

I UNISON 2005 ble proximity filter funnet å være et effektivt filter for å redusere tilfeldig støy. Med andre ord ekko som opptrer høyst sporadisk. Dataene fra Bjørnafjorden vil bli kjørt gjennom et proximity filter, og denne rapporten ser på hvilken effekt filtreringen vil ha for resultatene av målfølgingsalgoritmen.

For analyseøyemed er det av interesse å kunne eksportere den informasjonen målfølgeren legger til hver målfølgning. Filer for denne typen dataeksport blir utviklet og beskrevet.

Rapporten inneholder:

Kapitel 2: Bakgrunnsinformasjon

Kapitel 3: Arbeid gjort med målfølgning

Kapitel 4: Arbeid gjort med proximity filter

Kapitel 5: Konklusjon av arbeid



## 2 BAKGRUNNSTEORI

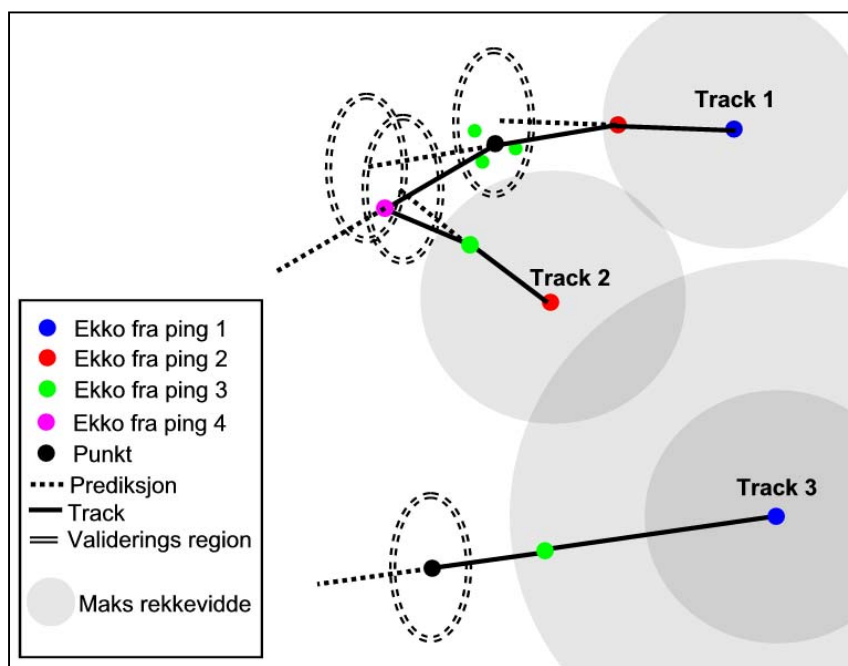
Uten å gå ned i for stort detaljnivå, vil dette kapitlet gi en teoretisk oppklaring rundt dataassosiasjon, flermodell interaksjon og proximity filter.

### 2.1 Målfølgeralgoritme

Teorien som er benyttet for å lage IMM-PDA filteret finnes fritt tilgjengelig i (1), (2) og (3). For å sette seg inn i selve programkoden vil rapporten til Løland, være til hjelp for å forstå fremgangsmåten. Hovedpunktene i teorien bli gjennomgått i dette kapitlet.

#### 2.1.1 Enkel forklaring av målfølgeralgoritmen

Målfølgeren er basert på teori om flermodell interaksjon (IMM). Det vil si modeller som kan hjelpe til å avgjøre om et mål er ekte eller falskt. Dette er kombinert med assosiasjon av målingene til eksisterende track, kalt sannsynlig data assosiasjon (PDA) (2). Her følger en enkel forklaring på hvordan målfølgeren arbeider.



Figur 2.1: Viser enkelt hvordan målfølgingsalgoritmen arbeider

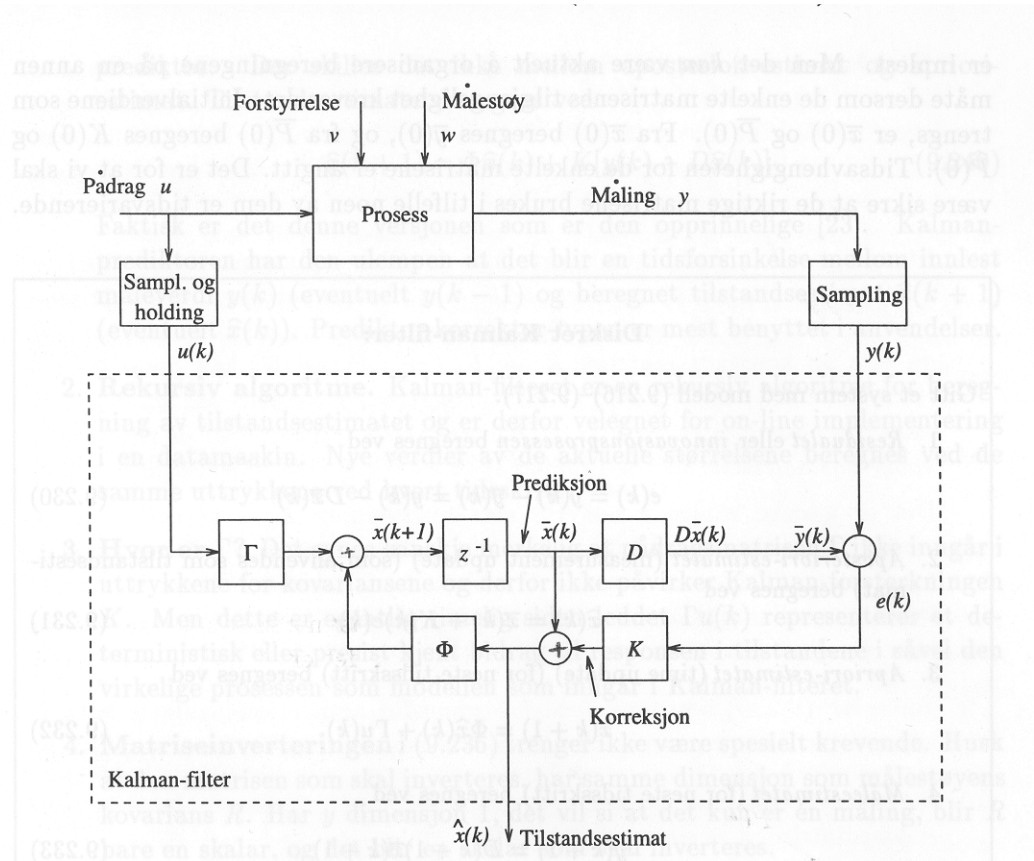
Når målfølgingsalgoritmen har ekko fra to ping, setter den opp en søkesirkel rundt de første ekkoene. Sirkelen har radius lik den maksimale antatte avstanden målet kan ha gått fra et ping til et annet. Ligger det et ekko fra neste ping i sirkelen, så blir et track opprettet mellom ekkoene (se track 1 og 2). Hvis det ikke kommer ekko innenfor sirkelen i det neste pinget, blir søkesirkelen fordoblet og ekko fra neste ping kan lage track med ekkoet, som da er to ping gammelt (se track 3). Hvis det ikke det finnes noen kandidater i den fordoblede søkesirkelen blir ekkoet i senter forlatt.

Når et track er opprettet, blir det beregnet en prediksjon om hvor neste ekko vil ligge hvis ekkoene skal stamme fra et ekte mål. For å tillate avvik vil det bli lagd en valideringsregion rundt den predikerte målingen. De ekko som befinner seg inne i dette området vil bli vektet og slått sammen til ett punkt som tracket koples til (se track 1, ekko fra ping 3). Ligger det ingen ekko i valideringsområdet vil prediksjonen bli brukt (se track 3).

Hvis to track bruker samme ekko blir det undersøkt om det ser ut til at vil sammenfalle. Hvis vinkelen mellom hastighetsvektoren til prediksjonen til hver av trackene ligger under en gitt grense, vil det yngste tracket bli avsluttet til fordel for det eldste (se track 3, ekko fra ping 4).

### 2.1.2 Kalmanfilter

Hvis man kan beskrive tilstandene i en prosess matematisk, kan man også prediktere prosessens måling i neste tidssteg. Et Kalmanfilter bruker denne prediksjonen for å evaluere prosessens måling. Slik kan filteret begrense innvirkningen av målestøy.



Figur 2.2: Prosess med Kalmanfilter (4)

Figuren forklarer hvor de forskjellige betegnelsene stammer i fra. Legg spesielt merke til forskjellen på prediksjon og estimat da disse vil bli hyppig referert til gjennom rapporten.

$\Gamma$	pådragsmatrise
$\Phi$	tilstandsmatrise
$D$	målematrise
$K$	Kalman-forsterkningen
$e$	innvasjonen
$\hat{x}$	estimat
$\bar{x}$	prediksjon
$y$	måling
$\Theta$	peiling
$r$	avstand

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} \Theta \\ r^{-1} \end{bmatrix}$$

$$\Phi = \begin{bmatrix} 1 & 0 & \Delta T_k & 0 \\ 0 & 1 & 0 & \Delta T_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \bar{\mathbf{x}} = \begin{bmatrix} p_x \\ p_y \\ v_x \\ v_y \end{bmatrix} = \begin{bmatrix} r^{-1} \\ \Theta \\ \dot{r}r^{-1} \\ \dot{\Theta} \end{bmatrix}$$

Legg merke til at målingen gir avstanden på inversform. Fra prosesseringskjeden til sonaren kommer avstand og peiling på normal form, men det er valgt å invertere avstanden i implementasjonen av målfølgingsalgoritmen. Erfaringer fra passiv sonar sier at målestøyen på avstanden øker eksponentielt når avstanden øker. Ved å oppgi avstanden på inversform vil usikkerheten får en jevnere fordeling. Det er i (5) antatt at dette også gjelder for aktiv sonar. I koden skifter peiling og avstand plass i forhold til prediksjon og måling.

I utregningen av prediksjonen har vi ikke tilgang til pådraget til målet og systemet er derfor ikke styrbart. Og det var forventet siden systemet skal observere mulige mål. Et krav til Kalmanfiltrering er at systemet er observerbart. For å teste det sier vi:

$$\text{rank} \left( \begin{bmatrix} \Phi \\ D\Phi^2 \end{bmatrix} \right) == n \quad \text{der } \Phi = n \times n - \text{matrise} \quad (2.1)$$

Systemet tatt fore her oppfyller dette kravet, og målingene kan Kalmanfiltreres.

### 2.1.3 Vekting av ekko

Målingene som er inne i valideringsområdet blir vektet etter hvor sannsynlig det er at de stammer fra målet som følges. Dette kalles ”*probabilistic data association*” (1), side 166. Hvert av de  $\mathbf{m}_k$  ekkoene får en vekting  $\beta$  etter formelen under.

$$e_i = P\mathbf{g}^{-1} |2\pi \mathbf{S}|^{-1/2} \mathbf{e}^{-1/2\mathbf{v}_i^T \mathbf{S}^{-1} \mathbf{v}_i} \approx P\mathbf{g}^{-1} (2\pi)^{-n/2} |\mathbf{S}|^{-1/2} \mathbf{e}^{-1/2\mathbf{v}_i^T \mathbf{S}^{-1} \mathbf{v}_i} \quad (2.2)$$

$$b = \frac{\mathbf{m}_k (1 - PdPg)}{PdPg\mathbf{A}_k} \quad (2.3)$$

$$\beta_i(k) = \frac{e_i}{b + \sum_{j=1}^{m_k} e_j}, \quad i = 1, \dots, m_k$$

$$\beta_0(k) = \frac{b}{b + \sum_{j=1}^{m_k} e_j} \quad (2.4)$$

- $\mathbf{v}_i$  innovasjonsvektoren (avviket mellom prediksjon og måling) til målingene
- $\mathbf{S}$  innovasjonsvariansen
- $n$  dimensjonen til målevektoren
- $\mathbf{A}$  valideringsregionens areal
- $Pd$  sannsynlighet for deteksjon av ekte mål
- $Pg$  sannsynligheten for at målingen faller i valideringsregionen

Prediksjonen får vekten  $\beta_0$ , mens de andre ekkoene får hver sin  $\beta_i$ . Estimaten om hvor målet befinner seg er gitt av formelen:

$$\hat{\mathbf{x}}_k = \sum_{a=1}^{m_k} (\beta_{k,a} \mathbf{y}_{k,a}) + \beta_0 \bar{\mathbf{x}}_k \quad (2.5)$$

$a$       validerte ekko  
 $k$       målnummer i track  
 $m_k$     antall ekko forbundet med track

Dette tilsier at man tar alle de vektete ekkoene og legger i sammen til ett estimat, men hvis det ikke finnes noen ekko i valideringsregionen blir estimaten lik prediksjonen.

#### 2.1.4 Vekting av track

Vektingen av track blir kalt ”*True Track Probability*” eller ”*sannsynlighet for ekte track*”. Den blir funnet ved å definere ulike modeller for et system. Metoden kalles *Interacting Multiple Model*. Modellene gis en sannsynlighet etter hvor godt målingene stemmer med de enkelte modellene. TTP-verdien til tracket vil da være lik sannsynligheten til den modellen som beskriver et ekte mål. Teoretisk skal det lages mange forskjellige modeller av systemet, og så avgjøre hvilken som passer best. Her er det først og fremst modeller for ekte mål (observerbart) og falskt mål (uobserverbart) som benyttes. Dette kan for eksempel føre til modellene:

$M_0$ : falskt mål  
 $M_1$ : ekte mål med konstant hastighet  
 $M_2$ : ekte mål som akselererer

I målfølgingsalgoritmen som analyseres i denne rapporten er det kun brukt de to første modellene. Sannsynligheten for å oppdage et ekte mål er gitt av  $Pd$ , mens sannsynligheten for å oppdage et falskt mål er  $Pd = 0$ . Det gir følgende funksjoner for å vekte modellene (2):

$$\Lambda_t = p[\mathbf{Z} | \mathbf{M}_t, \mathbf{Z}^{k-1}] = \mathbf{V}^{-m} \gamma_0 + \mathbf{V}^{-m+1} \sum_{j=1}^{m_k} [Pg^{-1} \cdot N[\mathbf{v}_j^t; 0, \mathbf{S}^t] \cdot \gamma_j[m_k]], \quad (2.6)$$

$$\gamma = \begin{cases} PdPg m_k^{-1} & j = 1, \dots, m_k \\ 1 - PdPg & j = 0 \end{cases}$$

$$\Lambda_1 = p[\mathbf{v}_1^1, \dots, \mathbf{v}_{m_k}^1 | \mathbf{Z}^{k-1}] = \left[ b + \sum_{j=1}^{m_k} e_j \right] \cdot \frac{PdPg}{m_k} \mathbf{V}^{-m_k+1}, \quad \text{observerbar}$$

$$\Lambda_2 = p[\mathbf{v}_1^2, \dots, \mathbf{v}_{m_k}^2 | \mathbf{Z}^{k-1}] = \mathbf{V}^{-m_k}, \quad \text{uobserverbar}$$

Etter at modellene er vektet kan man gi modellene sannsynligheten:

$$\mu^t = \frac{1}{c} \Lambda_t \bar{c}_t \quad (2.7)$$

der  $\bar{c}_t$  er sannsynligheten for at tracket kan skifte modell, og den verdien er utledet fra det historiske modellvalget til tracket og en Markov-”skiftmodell”-matrise.

## 2.2 Proximity Filter

Et proximity filter er et filter som setter krav til repeterende kontakter. Det vil si at et ekko slipper igjennom filteret kun hvis det har vært et ekko i området før.

I (6) er det gjennomført forsøk på ekkodata med et PF med et søkevindu på 25x25m, og en søkehistorie på 10 ping tilbake. Resultatene viser at et PF effektivt filtrerer ut ekko som opptrer sporadisk, men at man skal være kritisk til utfallet fordi mulige svake kontakter fra et ekte mål kan filtreres bort.

For målfølgingsalgoritmen vil filteret tenkes å ha noe av den samme effekten som parameteren for øvre tillatte hastighet. Et filter som beskrevet ovenfor vil gi en maksimal hastighet på 5 meter per sekund, gitt at det er 5 sekunder mellom oppdateringene. Det tilsvarer en hastighet på nesten 10 knot, mot 30 knot som er satt som maksimal tillatt hastighet i målfølgeralgoritmen.

## 2.3 Akronymer

IMM	Interacting Multiple Models
PDA	Probabilistic Data Association
TTP	True Track Probability
PF	Proximity Filter
SNR	Signal to Noise Ratio
Pd	Target Detection Probability
Pg	Gate Probability



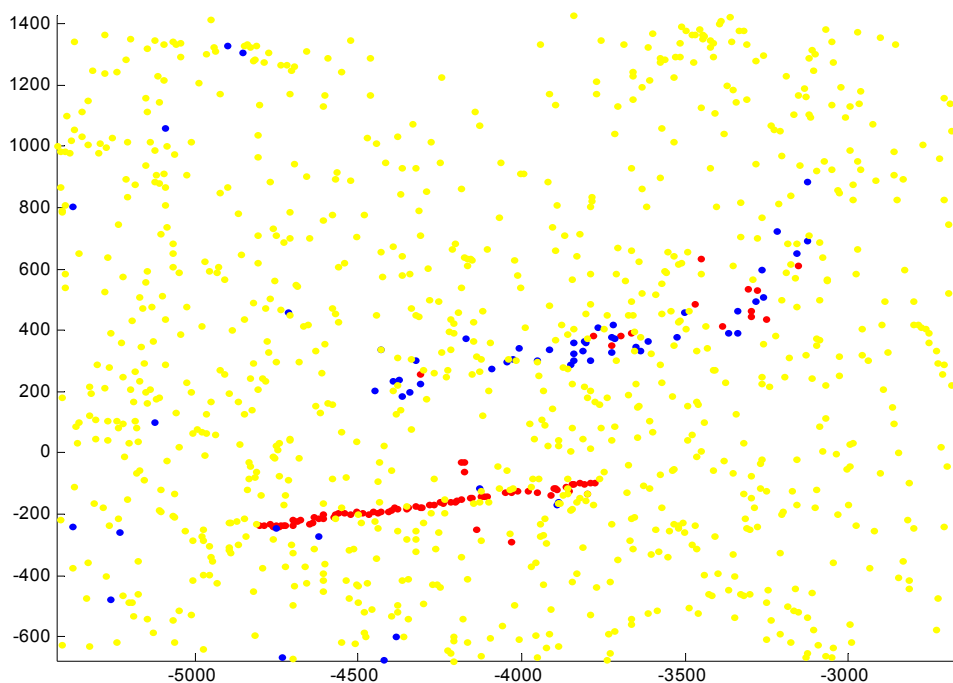


### 3 ARBEID MED MÅLFØLGINGSALGORITME

Det er allerede utviklet en MatLab-kode som utfører målfølgning ved hjelp av ”Interacting Multiple Model – Probabilistic Data Association” (IMMPDA). Måledatasettet fra (5) blir brukt som måldata i IMMPDA. Resultatet blir vist grafisk som mulige track (målbaner) der et mål kan ha gått. Arbeidet som blir beskrevet i dette kapittelet er studie av målfølgerens egenskaper. Hvordan målfølgningsteorien er implementert står beskrevet i (7). Det er gjort noen endringer i koden som har direkte innvirkning på målfølgningen. Disse er beskrevet i appendiks A. For å eksportere resultatene av målfølgningen er det lagd et filformat som er beskrevet i appendiks B.

#### 3.1 Parameterinnstillinger

Parametrene i målfølgingsalgoritmen er, med noen få unntak, relatert til sannsynligheter og støy. Disse parameterverdiene har vi et for lite grunnlag til å sette eksakt, og de må derfor fininnstilles. Risikoen med innstillingen er at vi spesialisere parametrene til å passe ett måledatasett, derfor må man også vite hvilken effekt parametrene har på målfølgningen. Vi ønsker oss en tracking som kan følge det virkelige målet i best mulig grad. Men på grunn av at det ikke er alltid vi har like klare ekko fra målet som i datasettet fra Bjørnafjorden, så vil multipath-ekkoene fra målet være et alternativt mål. Multipath-ekkoene oppstår når lyden går en alternativ rute mellom observatør og mål, som følge av refleksjon i overflate og havbunn. Disse ekkoene er svake og usikre. Et eksempel på multipath-ekko er vist i figur 3.1.



Figur 3.1: Utsnitt av ekkoplott med SNR-fargekode

Ekkoene i figuren er fargelagt etter signal/støy-forholdet. Signal med SNR på over 300 er merket med røde prikker, mens de med SNR mellom 150 og 300 er merket med blått og alle ekko med lavere SNR er gule. Som vi ser av figuren så er målets ekko veldig klare, og man har heller ikke problemer med å se ekkoene som gir en ”skygge” av målet på grunn av

flerveistransmisjon. Skyggen har langt større varians på målingene enn målet, og er derfor også vanskeligere å følge.

### 3.1.1 Grafer og tall som gir grunnlag for analyse

I dette delkapittelet vil grafene og tallene som gir grunnlag for analysen vil bli forklart. Grafer som blir gjort tilgjengelige i analysen av parameterinnstillingene er:

- oversiktbildet for å se hva som skjer med trackene med høyest TTP, og om noen av disse følger ekkoene til det ekte målet eller skyggen
- kurver av egenskapene til trackene i form av avstand og peiling mellom mål og observatør, og fart og kurs til målet
- fordelingen av sannsynlighet for ekte track og fordelingen av levetiden for alle trackene som oppstod under kjøringen, med lengde på over to. Det vil si track som har levd i over tre ping. Det blir vist hvordan klokkeformen på fordelingen endres med parametrene

Nærmere forklaring av de enkelte grafene blir forklart i videre i dette kapittelet.

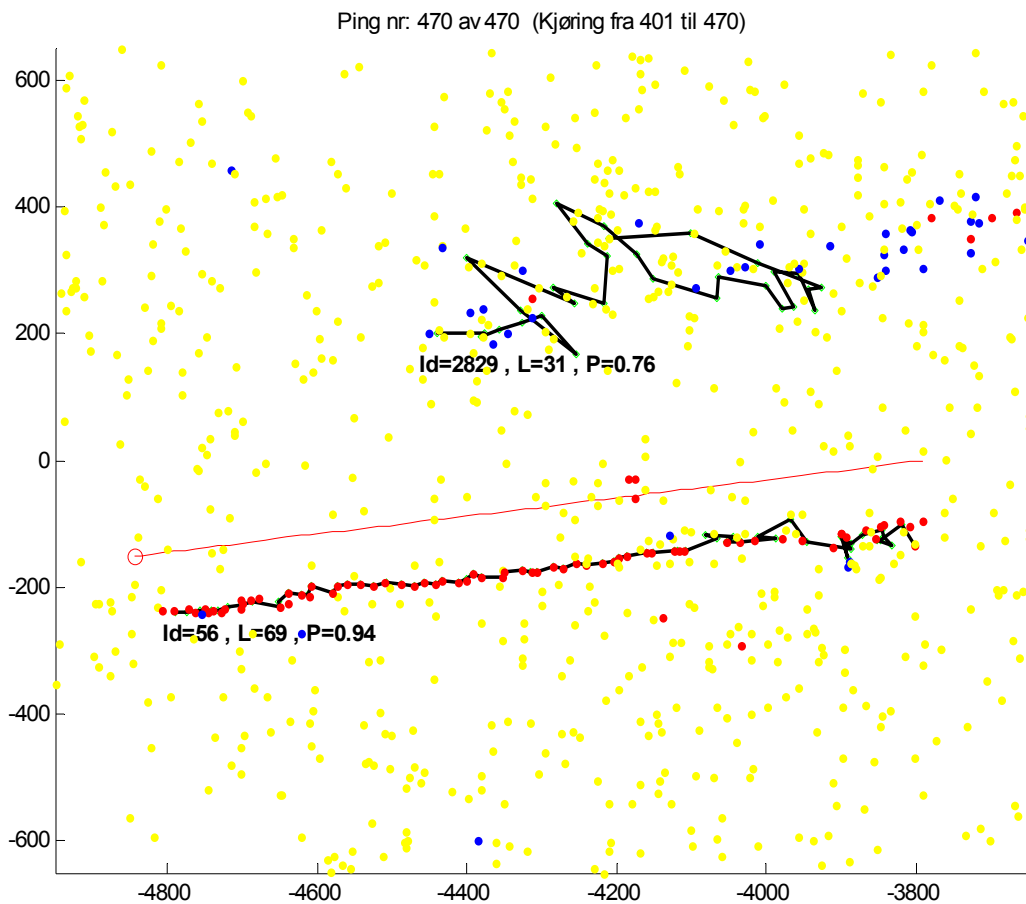
Parametrene som stilles er gitt i en oppstartsfil. Denne er gjengitt nedenfor. For å teste ut parametrene vil kun én parameter justeres av gangen. Oppsettet vist her er grunninnstillingen.

```
% Track deletion criterias
vt_min= 0;           %[m/s] Minimum speed for maneuvering
vt_max= 15;         % Approx 30knots [kts]=[m/s]/0.514[m/s]
my_min = 0.05;     % Minimum true track probability

% Probablistic data association settings
Pg = (1-0.01);     % Gate Probability
Chi_val = 9.21;    % Chi-value for 2 degrees of freedom
                % and Pg% confidence region, linked with Pg
Pd= 0.90;         % Target Detection probatibility

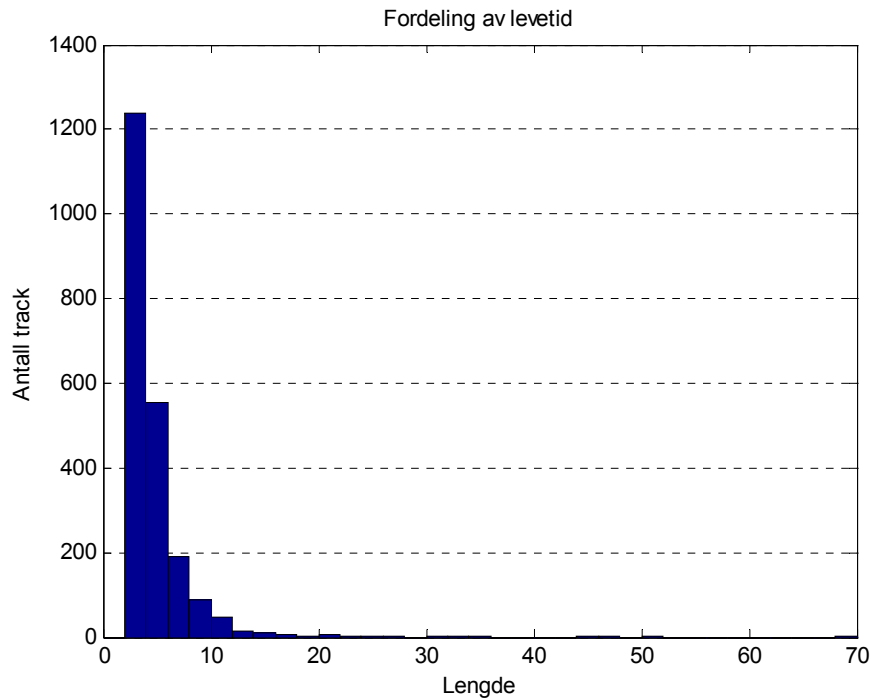
% System mode
r1=(0.5*pi/180)^2; % Measurement noise to bearing
r2=30^2;          % Standard deviation equal to 50-60m to
                % the range between meas.
r2_m=1.1e-11;     % Modified range variance.
                % Var when Inverse uniform(4200m,5000m) .
H=[0 1 0 0;1 0 0 0]; % Measurement matrix
R=[r1 0;0 r2_m];   % Measurement noise matrix

% Interacting multiple model settings
MPd2=[1-0.1 0.1;0.1 1-0.1]; % MARKOV CHAIN PROB.
```

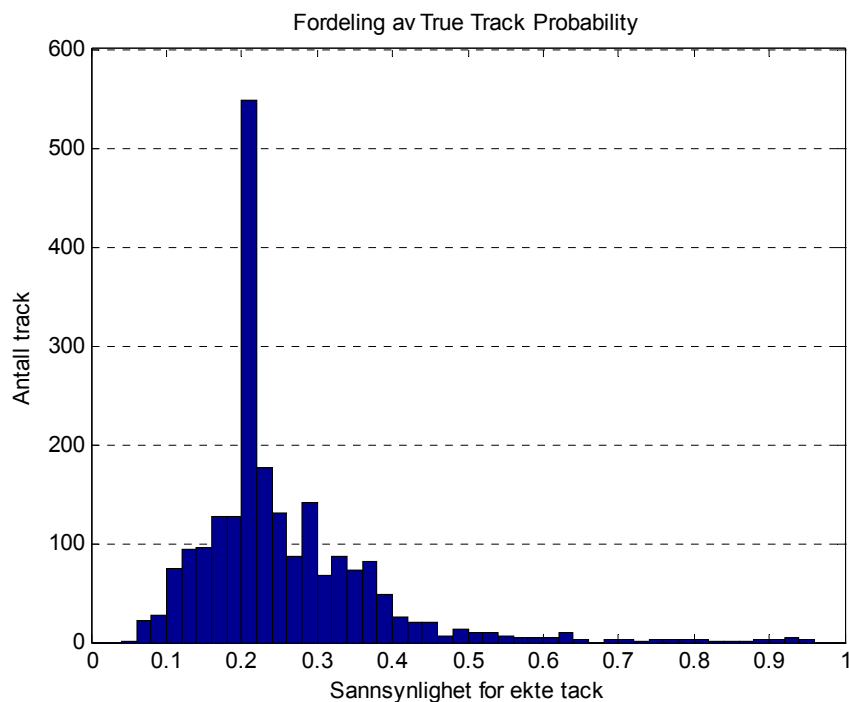


Figur 3.2: Viser track som kom ved første parameterinnstilling

I figur 3.2 ser vi at det kom to track. Enden der informasjonen om tracket står (Id=56, L=69, P=0,94), er siste målepunkt i tracket. Målet har altså en forflytning fra høyre til venstre i figuren. Informasjonen som er gitt er ID som er kjennemerket til tracket, L er levetiden og P er TTP. Den røde linjen er den loggførte til posisjonen til målet, gitt av GPS-korrigert treghetsnavigasjon. Det lengste tracket i figuren har en optimal lengde i forhold til hvor mange ping som ble kjørt, og sannsynligheten for at det er et ekte track er høy. Tracket som er oppstått ved multipathekkoene er under halvparten så langt og har en lavere TTP.



Figur 3.3: Levetidene ved første parameterinnstilling



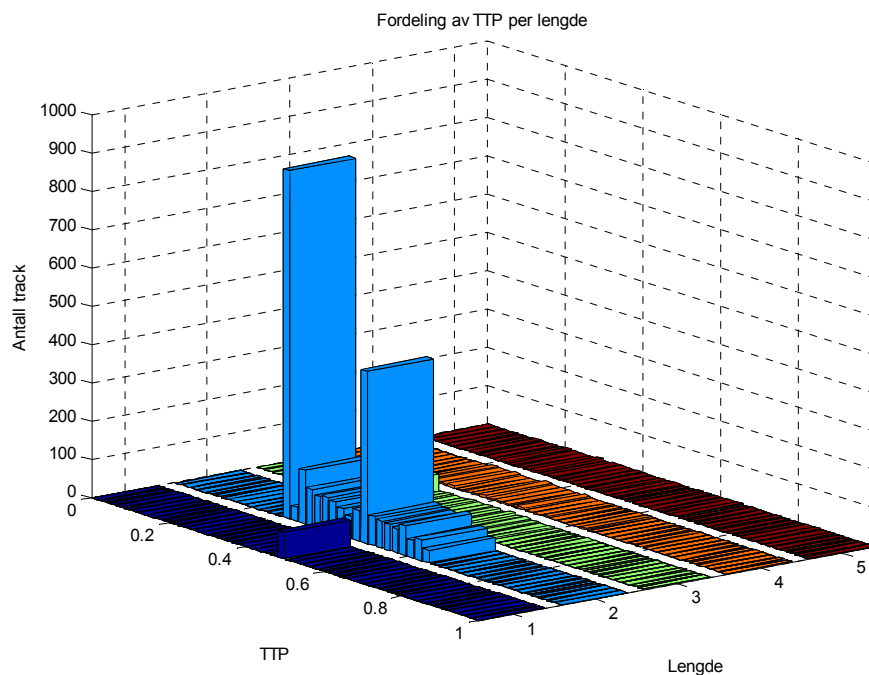
Figur 3.4: Sannsynligheten for track ved første parameterinnstilling

Etter at kjøringen fra ping 401 – 470 er fullendt, plottes dataene for alle track, eksisterende ved ping 470 eller som er slettet underveis. Track med lengde én eller to er ikke tatt med i plottene. Disse trackene er i oppstartsfasen, og der vil flesteparten av trackene også ende. Det er derfor valgt å se på track med en levetid på over to ping.

I figur 3.3 kan man se fordelingen av levetider på track. Søylene i histogrammet er delt inn i steg på to i x-aksen. Som figur 3.3, viser så dør de fleste trackene før de er blitt 10 ping lange. Man ser også tracket som følger det ekte målet med en lengde  $\in [69, 70]$ .

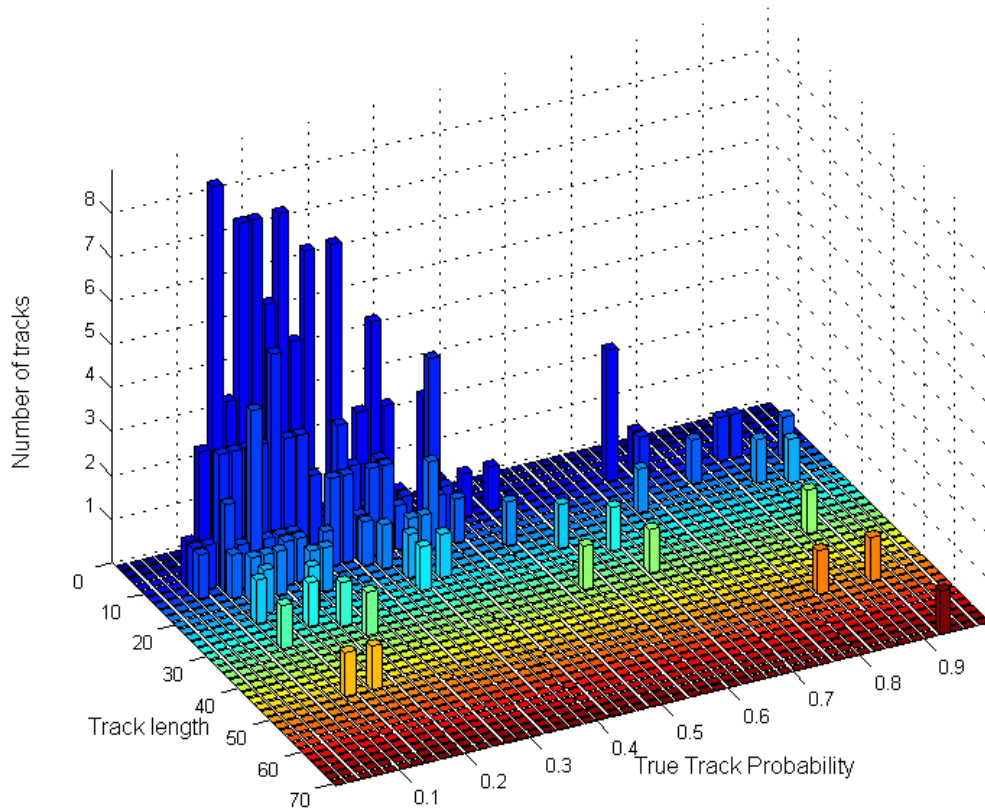
I figur 3.4 er det fordelingen av sannsynligheten for ekte track som vises, der  $TTP \in [0, 1]$ , og ved  $TTP > 0.5$  er det større sannsynlighet for at målet er ekte enn falskt. Det er gjennomsnittet av de tre siste TTP-verdiene til trackene som brukes når tracket er over 3 ping langt. Det er gjort

fordi TTP svinger noen prosent mellom oppdateringene, og en gjennomsnittsverdi kan fortelle hvilket område verdien svinger rundt. I figur 3.5 ser man at en stor andel av trackene har  $TTP \in [0.29 \ 0.3]$ . Dette har trolig sin forklaring i at alle track starter med  $TTP = 50\%$ , og i de tilfeller tracket blir slettet kort etter at det er opprettet ser det ut til at den gjennomsnittlige TTP bli 30%. Ser man på fordelingen av TTP i forhold til lengde, vist i figur 3.5, ser man at det er track med lengde under 3 som oftest har verdiene 0.5 og 0.3. Kjøring av koden i avlusningsmodus viser at track som har lengde på to og ikke har noen ekko i sitt valideringsområde vil få en TTP på 30% på grunn av at kun prediksjonen blir brukt. Et ekte mål forventes å gi målinger, og et falskt forventes ikke å gi målinger. En målfølging der målingene opphører slik at prediksjonen må brukes tyder på et modellskifte fra ekte til falskt mål. Resultatet er at sannsynligheten for ekte mål reduseres.



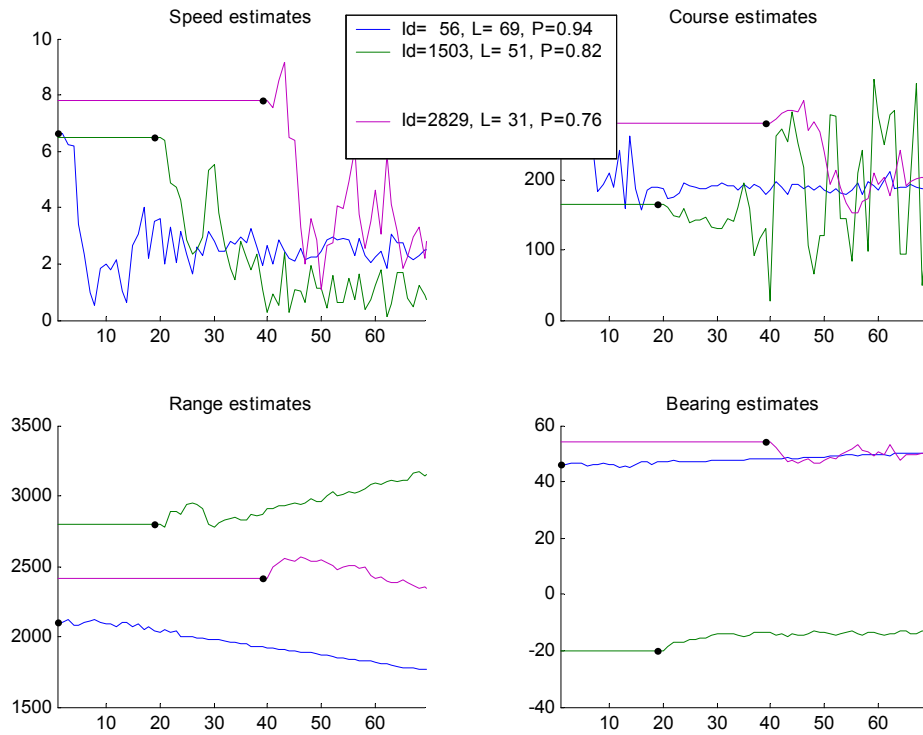
Figur 3.5: Fordeling av TTP i forhold til lengde fra 1 til 5

Figur 3.6 viser de track som har lengde på over 4. Man kan se at fordelingen er mer spredd, og at det ikke er noen spesielle TTP-verdier som skiller seg ut i så stor grad som i tilfellene der lengden er på to ping. Det ekte målet skiller seg ut som en rød søyle helt til høyre i figuren, mens tracket på skyggen er mindre tydelig i figuren.



Figur 3.6: Fordeling av TTP i forhold til lengde fra 5 til 69

I målfølgingsalgoritmen ligger det en modell som beskriver egenskapene for et mål. En slik modell benyttes til å prediktere hvor man i neste ping vil få et ekko fra målet. Det forutsetter at man har en modell som kan beskrive dynamikken til målet best mulig. Når man får nye målinger, vil det kunne være flere kandidater som kan være korrekte. De blir derfor vektet og til sammen utgjør de et estimat på hvor målet befinner seg. Ved å se på kurvene for estimatene i figur 3.7 kan man se om egenskapene til målet som følges kan være et ekte mål. Da vil egenskapene stabilisere seg slik at man vil kunne se en trend. Ser man på kurvene for kursestimat, så er estimatet til tracket som følger det ekte målet, track 56, stabilt. Tracket som følger ”skyggen” av målet, track 2829, varierer noe mer. Track 1503 er et falskt track, og som man kan se så svinger kursen mye. Men på grunn av målets lave hastighet, og usikkerhet i målinger, blir denne svingningen tolerert. I de andre grafene skiller det falske tracket seg mindre ut.



Figur 3.7: Estimater forbundet med track. Blått: virkelig mål. Rødt: flerveisrefleksjonen til virkelig mål. Grønn: track med opphav i topografisk struktur

### 3.1.2 Innvirkning av Target Detection Probability

Target Detection Probability ( $P_d$ ) inngår i funksjonene for beregning av vekting av ekko, og modellsannsynlighet. Man ønsker seg et system som best mulig skiller ut et ekte mål fra en mengde av mulige mål, og  $P_d$  har innvirkning på dette.

For å få en bedre forståelse av innvirkningen av  $P_d$  i målfølgeralgoritmen er det gjort kjøring der  $P_d$  har hatt verdiene 60%, 75%, 80%, 90%, 95% og 99%. Av teorien ser man at en høy verdi på  $P_d$  vil gi en høy vekting av ekkoene som faller inn i valideringsregionen (figur 2.1), men lav  $P_d$  vil gi en høy vektig av prediksjonen. I tabellene 3.1 – 3.3 kan man se tre eksempler på utfallet av vektigen av ekko i valideringsregionen ved høy og lav  $P_d$ .

Pd=99%	$\bar{y}$	1,9 %
	y	98,1 %
Pd=75%	$\bar{y}$	24,6 %
	y	75,4 %
Avvik	$e_x$	53,2 m
	$e_y$	111,4 m
	$\sqrt{e_x^2 + e_y^2}$	123,5 m

Forklaring:

$\bar{y}$  : prediktert måling

y : ekkopunkt i valideringsregion

e : avviket mellom prediksjon og ekkopunkt

Tabell 3.1

Pd=99%	$\bar{y}$	6,3 %			
	y	5,9 %	29,8 %	30 %	29 %
Pd=75%	$\bar{y}$	53,6 %			
	y	2,6 %	14,8 %	14,6 %	14,4 %
Avvik	$e_x$	256,5 m	3,6 m	2,6 m	2,6 m
	$e_y$	345,8 m	220,8 m	221,8 m	222,8 m
	$\sqrt{e_x^2 + e_y^2}$	430,5 m	220,8 m	221,8 m	222,8 m

Tabell 3.2

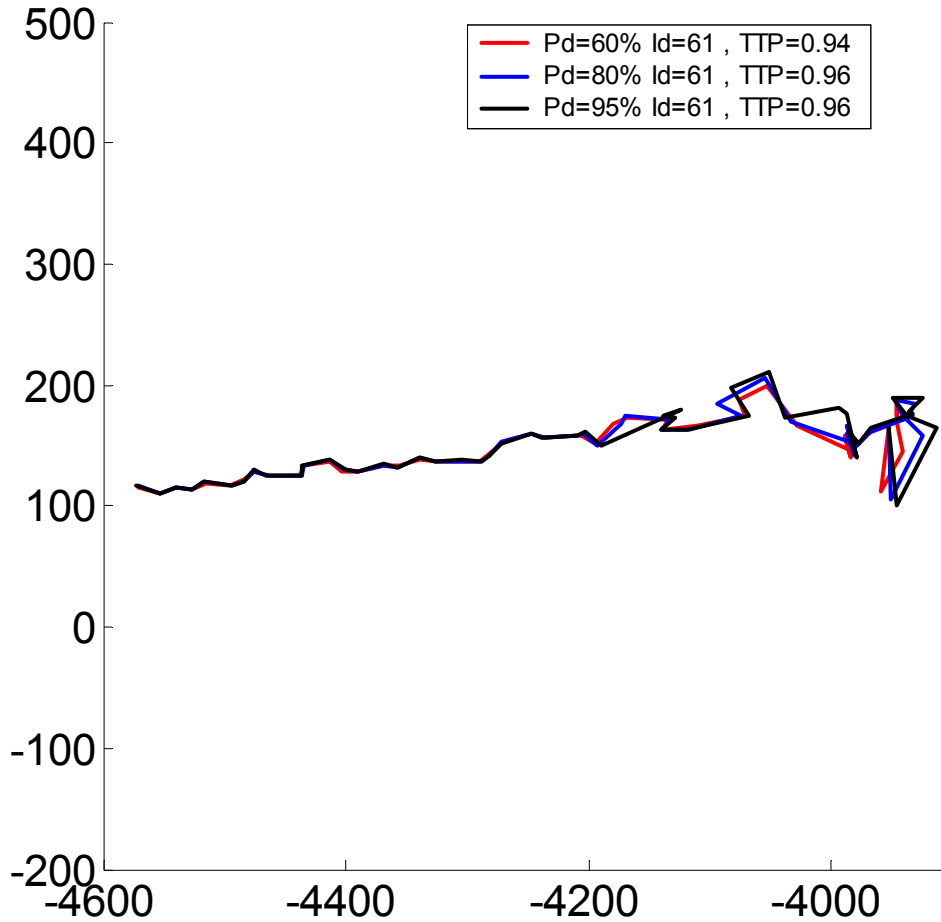
Pd=99%	$\bar{y}$	1,5 %	
	y	2,6 %	95,8 %
Pd=75%	$\bar{y}$	21 %	
	y	2,1 %	76,8 %
Avvik	$e_x$	188,3 m	60,7 m
	$e_y$	16,3 m	57,3 m
	$\sqrt{e_x^2 + e_y^2}$	189,0 m	83,5 m

Tabell 3.3

Tabellene viser vektingen av ekko på tre forskjellige track, og hvert eksempel har ulikt antall ekko i valideringsregionen. Tabellene viser også avvikene mellom prediksjon og ekkopunkter. Som man ser gir et absolutt avvik på rundt 100m en høy vekting, men man må huske på at vektingen er også sett i forhold til forrige avvik. Dermed kan det være at 100 meters avvik er lite i forhold til hva man hadde i forrige ping, og gir derfor høy vekt til ekkoet i inneværende ping. Så sammenhengen mellom avvik og vekting er derfor ikke bare basert på tallene i tabellen, men likevel gir tabellen en indikasjon på hvilke avvik man har i målfølgeren.

En av følgene ved lav Pd er at trackene vil opptre stødigere enn ved høy Pd, men TTP-verdiene vil bli noe lavere. Grunnen til at tracket bli stødigere er at man, ved å vekte prediksjonen høyere, vil fortsette mer i samme retning som ved siste oppdatering. Man må derfor avgjøre hva man er ute etter. Ønsker man track som visuelt sett opptre noe mer som et ekte mål, så kan man sette Pd til en lav verdi slik at track-banene blir glattere. En følge av dette er flattere spredning av de forskjellige trackenes TTP (figur 3.9), og dermed vil det finnes flere track med TTP lik det ekte målet. Det blir da vanskeligere å skille ut det ekte målet. Setter man derimot en høy Pd vil tracket gå fra ett ekkopunkt til et annet. Siden man har usikkerhet i peiling og avstand vil banen opptre mer hakkete. Forskjellen kan sees i figur 3.8. Der ser man banen til det ekte målet med forskjellige verdier av Pd.

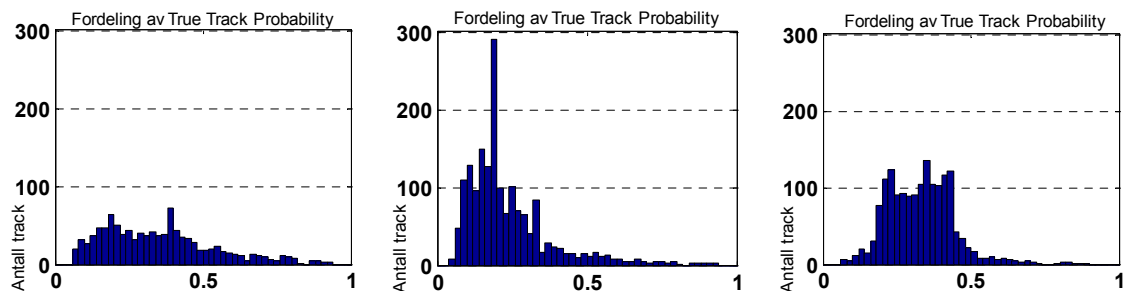




Figur 3.8: Det samme tracket ved forskjellige verdier av Pd

Som man kan se av figuren er det begrenset hvor glattet trackene blir ved lave verdier av Pd. En alternativ måte å glatte på, er å gjøre linjene tykkere, men det er en avgjørelse man må gjøre i arbeidet med visuell fremvising av dataene.

Det viser seg at antall track med lengde over 2 ping øker ettersom man øker Pd. Lave verdier på Pd gir færre track, men en bredere fordeling av TTP-verdier og høyere gjennomsnittslengde på trackene. Gjennomsnittet av fordelingen blir noe høyere når Pd økes, men ikke oppsiktsvekkende. Figurene nedenfor viser fordelingen av TTP etter noen av kjøringene.



Figur 3.9: Fordeling av TTP ved Pd lik 60%, 80% og 99%

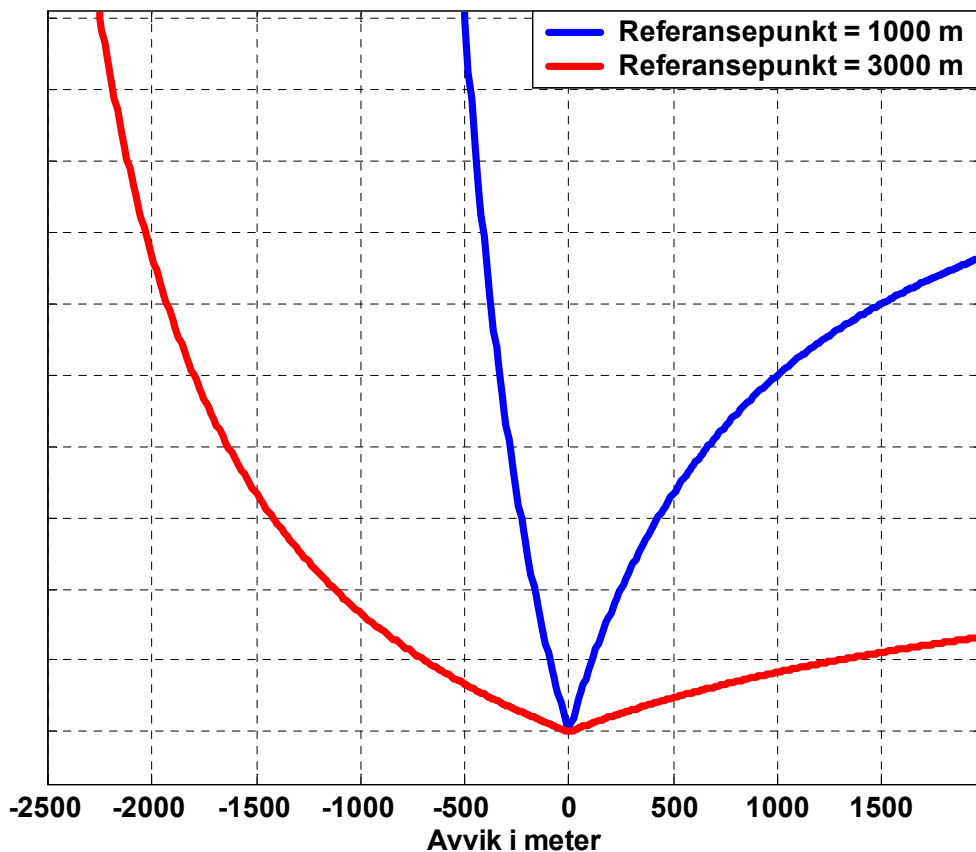
Spredningen synes være best med verdier rundt Pd = 90%. Da er fordelingen slik at det er få track som har TTP over 50%, men likevel er det ett eller flere som har TTP på over 90%.

### 3.1.3 Valideringsregion og innvirkning av *Gate Probability*

Parameteren *Gate Probability* ( $P_g$ ) er sannsynligheten for at ekkoet fra målet vil ligge i valideringsområdet (*Validation Gate*). Formen på området avhenger av avviket mellom prediksjon og måling i peiling og avstand. Størrelsen på området avhenger av verdien til  $P_g$ .  $P_g$  gir  $\gamma$  – som er oppgitt i en tabell for Chi-Square fordeling, og  $\gamma$  er den øvre grense i området. Dette er vist nedenfor i formelen for utregning av valideringsområdets mengde:

$$\mathcal{V}_{(k+1, \gamma)} = \{[\mathbf{y} - \bar{\mathbf{y}}_{(k+1|k)}]^T \mathbf{S}_{(k+1)}^{-1} [\mathbf{y} - \bar{\mathbf{y}}_{(k+1|k)}] \leq \gamma\} \quad (3.1)$$

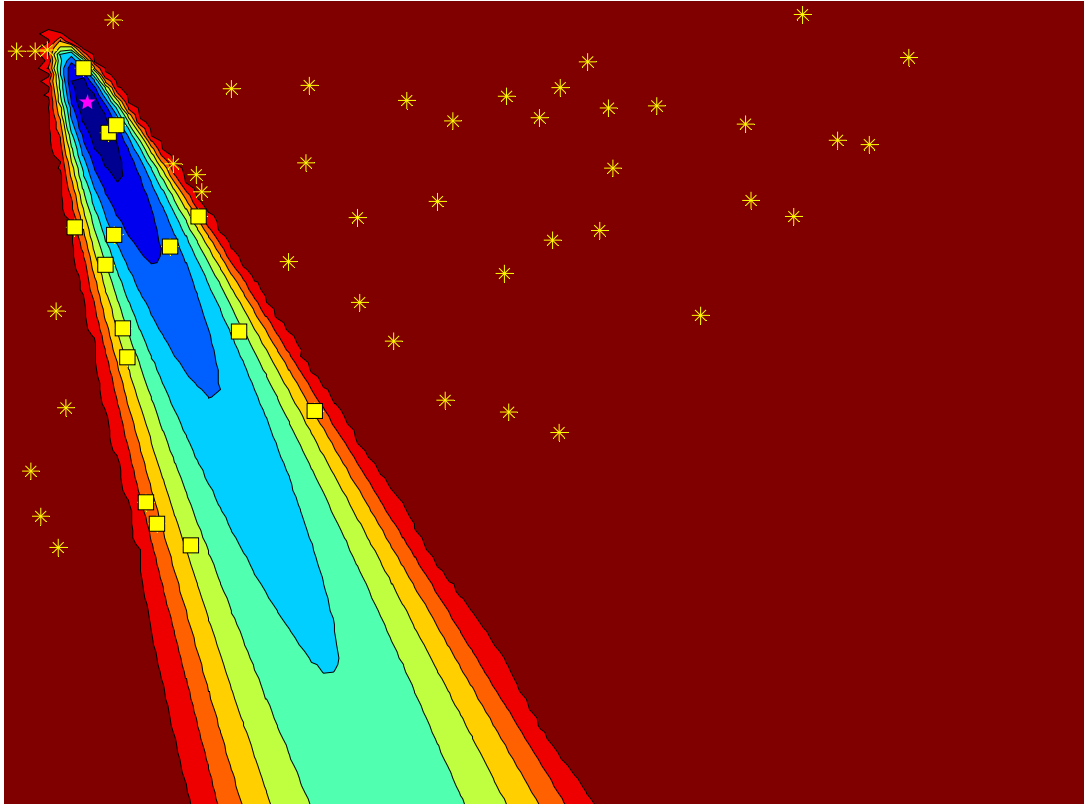
Når verdien til  $V_k$  er mindre enn  $\gamma$ , vil punktet ligge innenfor valideringsregionen.  $\mathbf{S}$  er måleprediksjonens kovarians, med andre ord en indikasjon for hvor korrekt forrige prediksjon var i forhold til estimatet. For å illustrere valideringsregionen ser man på avvikene som ligger til grunne for området. Hvis man har lineære avvik vil valideringsregionen ha en ellipse form. Man tillater da et like stort avvik i fremkant som i bakkant av prediksjonen. Målfølgingsalgoritmen som studeres i denne rapporten er basert på målinger i modifiserte polare koordinater. Disse gir et ulineært avvik i avstand, som vist i figur 3.10.



Figur 3.10: Eksempler på avvik i avstand i modifiserte polare koordinater

Figuren viser eksempler på avvik rundt to forskjellige referansepunkt. Den horisontale akse viser avviket i avstand. Ved et lineært avvik hadde verdiene på aksene vært like, men her er den vertikale akse avviket mellom observatør og referansepunkt blir oppgitt på invers form. Følgelig vil avviket gå mot uendelig når man nærmer seg observatøren, og vises ved at avvikskurvene er langt steilere i fremkant av referansepunktet. Man ser også at det tillates større avvik dess lengre man forflytter seg vekk fra observatøren. Ved formingen av valideringsregionen settes det en avviksgrense vha  $\mathbf{S}$ . Man kan tyde at en lav avviksgrense (y-

akse på figur 3.10) vil gi en tilnærmet ellipseform fordi det er nesten like langt i fremkant som i bakkant av referansepunktet, men en høy grense tillater nye målepunkt langt i bakkant av prediksjonen. Det vil gi en strekt ellipse. I figur 3.11 er det et eksempel på en slik strekt valideringsregion. Vi har her en prediksjon som skal det skal koples ekkopunkter til. Bildet er generert ut fra virkelige data fra målfølgingsalgoritmen.



Figur 3.11: Valideringsregionen i praksis.

Prediksjonen er markert med en rosa stjerne. Gule kvadrater og stjerner er ekko som ble registrert henholdsvis i og utenfor valideringsregionen. Valideringsregionen er fargelagt etter verdien av  $V$  i området, og alle verdier over  $\gamma$  er satt lik  $\gamma$ . Prediksjonen ligger her 833 meter fra observatøren, som ligger ute av figuren i retning av øvre venstre hjørne. Sonarrekkevidden har her vært på 4000 meter, og blir antydnet av buen av ekko diagonalt i bildet. På grunn av de modifiserte polare koordinatene blir valideringsregionen så stor at alle ekkopunkt bak prediksjonen blir dekket. I algoritmen er det kompensert for dette ved å betrakte kun de fire beste kandidatene i valideringsregionen. Behovet for en slik kompensasjon antyder at på grunn av de modifiserte polare koordinatene, får vi ikke den valideringsregionen vi ønsker når de tillatte avvikene er store. Et resultat kan tenkes å være at trackene vil ha en tendens til å trekke seg vekk fra observatør fordi det er ekko i bakkant av prediksjonen som faller inn i regionen. Denne hypotesen kan ikke bekreftes ut fra trackene som dannes i målfølgeren i dag uten å ha noe å sammenlikne med. Men hvis man forestiller seg en valideringsregion formet som en ellipse i figur 3.11, med den rosa stjernen i senter, så ser vi at flere ekko i fremkant ville ligge innenfor valideringsregionen. Resultatet ville være et estimat som lå nærmere observatøren, og kanskje gitt mindre avvik mellom prediksjon og estimat. En annen svakhet med den valideringsregionen vi har med de modifiserte polare koordinatene er at forholdet mellom avviket i peiling og avstand på en aktiv sonar blir feil. Avstandsmålingen på en aktiv sonar betinger at man bruker riktig lydshastighet, men denne usikkerheten er liten i forhold til usikkerheten i peiling. Begge vil økes proporsjonalt med avstanden. Ellipsen skulle derfor vært

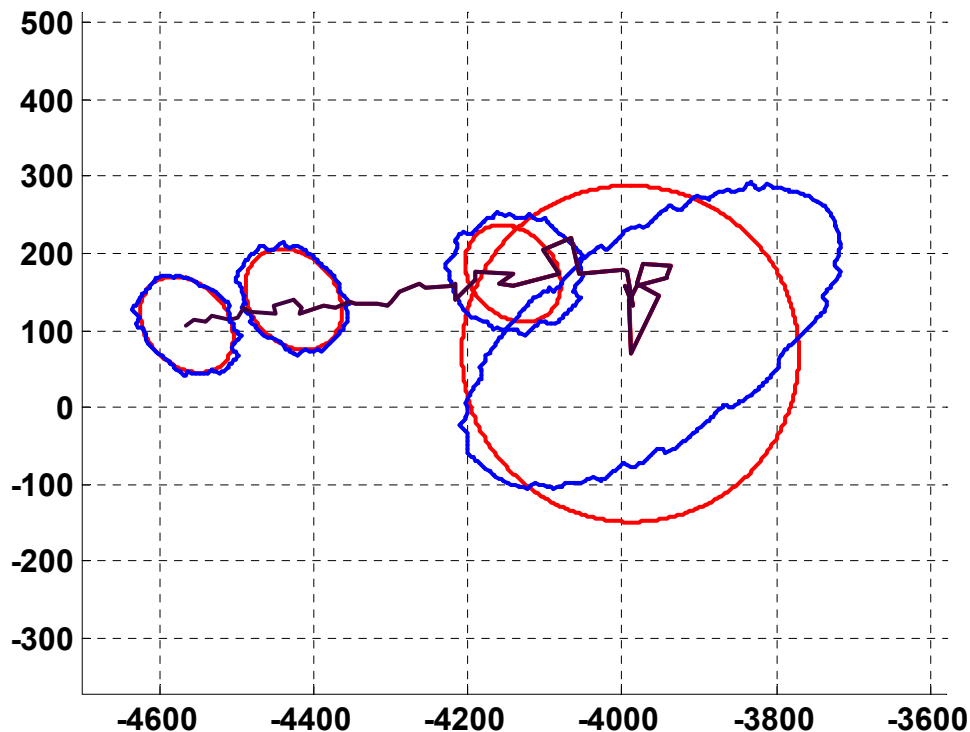
på sitt bredeste i peiling, altså den lengste radien skulle stått normalt på avstandsvektoren. Det er ikke tilfellet her. Det kan til dels skyldes at det kan være satt feil usikkerhet i peiling og avstand, men bortforklarer likevel ikke den uheldige konsekvensen ved å bruke den inverse avstanden som tilstandsvariabel.

Som nevnt gir  $P_g$  verdien av  $\gamma$ . Setter man en lavere sannsynlighet  $P_g$  vil  $\gamma$  også bli lavere. Slike lavere verdier kan man se i konturlinjene til valideringsregionen i figur 3.11.  $\gamma$  er i eksemplet lik 9, og som man ser er det ni konturlinjer.  $P_g$  verdier lik 90%, 95% og 97,5% gir  $\gamma$  lik 4.6, 6 og 7.4 etter Chi-square fordelingen. Det er da relativt enkelt å danne seg et bilde av hvordan  $P_g$  påvirker størrelsen av valideringsregionen.

Når man har lineært avvik, er det vanlig å tegne valideringsregionen ved å bruke formelen til en ellipse fremfor å teste mange forskjellige avvik. Man tar da egenverdiene til  $S$ , som betegnes  $\lambda$  og er lik variansen i peiling og avstand ( $\sigma^2$ ). Disse brukes i formelen for en ellipse:

$$V_{(k+1, \gamma)} = \sum_{i=1}^2 \frac{[y - \bar{y}^{(k+1|k)}]^2}{\lambda_i} = \gamma \quad \leftrightarrow \quad \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (3.2)$$

Sammenlikning mellom ellipsen og det virkelige området i figur 3.12 viser at likheten mellom ellipsen og valideringsregionen er best når tracket har små avvik mellom prediksjon og måling. Prediksjonen er basert på en modell for rettlinjett bevegelse, og derfor vil avviket være lite der tracket viser denne bevegelsen. Det kan vi se at tracket gjør til venstre i figur 3.12 (sort linje).

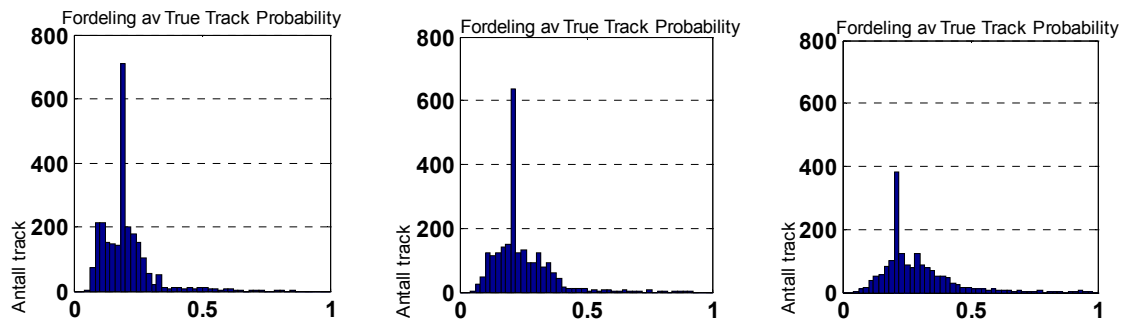


Figur 3.12: Valideringsregioner. Blå – virkelig Rød – basert på ellipse

Man ser også ut av figur 3.12 hvordan valideringsregionen skifter form etter hvert som nye målepunkter skal evalueres. Ved et tilfelle der man har et track som følger kravet om rettlinjett bevegelse, vil området reduseres slik at færrest mulig ekko skal forbindes til tracket.

Endring i  $P_g$  vil, i likhet med  $P_d$ , påvirke hvilken vektning ekkoene og trackene får. Fremgangen for å teste forskjellige verdier av  $P_g$  var er lik som ved  $P_d$ .  $P_d$  ble stilt tilbake til 90%, og verdiene 90%, 95%, 97,5%, 99%, 99,5%, 99,9% og 99,99% ble testet i  $P_g$  med tilhørende Chi-verdier.

Følgene av økning i  $P_g$  er at track får generelt lengre levetid, og som en følge blir færre track startet totalt da det allerede er track forbundet med de fleste ekko fra før. Av figur 3.13 kan man se en større spredning i fordelingen av TTP når  $P_g$  økes, og en svak forbedring i den gjennomsnittlige TTP. Best innstilling av  $P_g$  synes å være mellom 97,5% og 99,9%.



Figur 3.13: Fordelingen av TTP ved  $P_g$  lik 90%, 97.5% og 99.99%

Som en følge av bredere spredning av TTP ved høyere  $P_g$ , blir TTP-verdien til det ekte målet også høyere, men det blir også TTP-verdien til falske track. Hvis ekkoene fra et ekte mål har likt signal som den ekte målet i testdataene så kan  $P_g$  med fordel settes til 97,5% - 99%. Hvis signalet er likt flerveisrefleksjonen som opptrer i testdataene, da burde man sette  $P_g$  høyere slik at det ekte målet får tilnærmet lik TTP som de kraftigste falske trackene.

### 3.1.4 Innvirkning av tillatt hastighet

Målfølgingsalgoritmen sletter track hvis de bryter med rammen for tillatt absolutt hastighet. Den laveste tillatte hastigheten er vanligvis lik null, slik kan stillestående mål også bli detektert. Den øvre kan ligge på en verdi som er antatt å være den maksimale hastigheten til den typen mål man vil holde utkikk etter. Ved å redusere det tillatte hastighetsområdet vil færre track tilfredstille kravene, og arbeidslasten blir også redusert. Parametrens virkning er lett å forstå, og undertegnede velger ikke å kjøre tester med forskjellige innstillinger.

### 3.1.5 Innvirkningen av målestøy

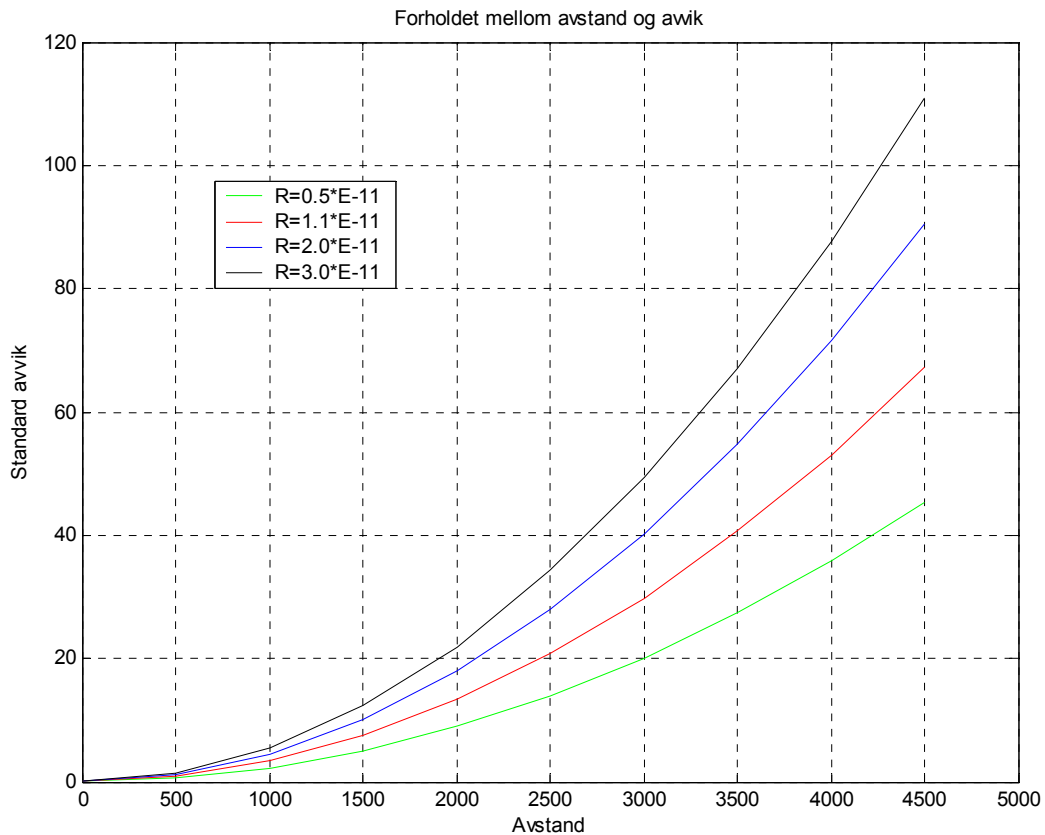
I modellen av et ekte mål er det en målestøymatrise  $R$  som er statisk når et track er 10 ping eller yngre. Opp til og med ping 10 er

$$R = \begin{bmatrix} \left(\frac{0.5\pi}{180}\right)^2 & 0 \\ 0 & 1.1 \cdot 10^{-11} \end{bmatrix}$$

Dette skal tilsvare støy på 0.5 grader i peiling, og avstandsmålingens støy er antatt å være avstandsavhengig, alternativt 30 meter ved initiering av track. Avstandsavhengighet vil som kjent si at man antar at målestøyen blir større når avstanden øker, og omvendt. Formelen for standard avvik ved gitt avstand er (5):

$$\sigma_{r_k} | r_k = \sqrt{R_{r_k} \cdot (r_k)^4} \quad (3.3)$$

Forskjellige verdien av R gir forhold mellom avstand og avvik som vist i figur 3.14.



Figur 3.14: Forhold mellom avstand og standard avvik

Av figuren ser vi hvordan det forventede avviket øker med avstanden. Kurvene har eksponentiale egenskaper.

Etter 10. ping er

$$R = R \cdot \left[ \frac{4.29}{\log(snr)} \right]^{1/2} \quad (3.4)$$

R for altså en korleksjon i målestøy med hensyn på SNR-verdien (signal-støy forhold) til målingene. Grunnen til denne løsningen er uviss.

For å teste innvirkningen av målestøy ble variabelen R, fra figur 3.14, variert mellom de oppgitte verdiene i figuren. De andre støyvariablene var satt til standard verdier. Endring av ga valideringsregioner som er flatere i avstandsretningen, men med store tillatte avvik er det ingen synlig forskjell. Egenskapene til track på ekte mål hadde ingen merkbare forandringer. Den faste avviksvariablen til avstanden som er  $\sigma_r^2 = 30^2$ . Denne blir kun brukt ved opprettelsen av nye track. Der er betingelsen av for at et mulig track skal kunne oppstå må to ekko ligge slik i forhold til hverandre at de oppfyller kravet:

$$|ekko(k) - ekko(k-1)| < \Delta T \cdot v_{\max} + 3\sqrt{\sigma_r^2} \quad (3.5)$$

Hvorfor uttrykket inneholder multiplikatoren 3 vites ikke. Vi ser at avviket blir  $\pm 90$  meter, og endring i parameteren vil kunne endre antall track som blir tatt med i et sett hypotetiske track for videre analyse i målfølgeren. Uttrykket inneholder en utvidelse av søkesirkelen som er basert på usikkerheten i avstand. Denne blir lagt på i alle retninger, selv i de retningene der kun usikkerheten i peiling har påvirkning. Det er derfor ikke studert forskjellige verdier av  $\sigma_r^2$ . Siste parameter i målestøy-matrisen er avvik i peiling. Sonaren er delt opp i 36 sektorer, det vil si på  $10^\circ$  hver. Maksimal avvik kan da være  $\pm 5^\circ$  på ekko med lav SNR-verdi. Det ble testet verdier fra  $0^\circ$  til  $5^\circ$  i  $R(1,1)$ . Resultatene viste at et stort avvik ga et høyt antall track som hadde fast avstand, men forskjellig peiling i forhold til observatøren. Valideringsregionen endret seg i forhold til innstillingen av peilestøyen.

På grunn av endringen i valideringsregionen kan man si at målestøy-matrisen påvirker målfølgingsalgoritmen som forventet. Støy i avstandsmålingen blir antatt å ha eksponentielle egenskaper, men for aktiv sonar er det nærliggende å anta at den er høyst proporsjonal med avstanden. SNR er tatt med i målestøyen, men i hvor stor grad den innvirker på støyverdien er usikkert. Det anbefales derfor å studere utnyttelsen av informasjonen som ligger i SNR-verdiene til å bestemme måleusikkerhet.

### 3.1.6 Innvirkning av Markov chain probability

Markov-matrisen gir en sannsynlighet for at et mål kan skifte fra falsk til ekte eller omvendt, gitt at det er de to eneste modellene. Dette er for å gi mulighet for at et mål kan bli "borte", for så å komme igjen noe senere. I teoretiske eksempler har Markov-matrisen for to modeller følgende utseende:

$$M_I = \begin{bmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{bmatrix}$$

Mens i målfølgingsalgoritmen benytter:

$$M_{II} = \begin{bmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{bmatrix}$$

Følgene av å bruke sistnevnte matrise er en strengere vurdering av trackene. Et tidssteg der det ikke kommer ekko i valideringsregionen vil ramme TTP-verdien hardere ved bruk av  $M_{II}$  enn ved bruk av  $M_I$ .

## 3.2 Forslag til videre arbeid på målfølgingsalgoritmen

Målfølgingsalgoritmen inneholder løsninger som kunne vært gjort annerledes. Forslag til endringer kommenteres her.

### 3.2.1 Tilstandsvariable

Det har vist seg at antagelsen av at avviket i avstand er avstandsavhengig gir uheldige utfall. Spesielt med tanke på størrelsen av valideringsregionen. I de verste tilfellene faller halvparten av ekkoene inn i regionen som følge av at tilstandsvariabelen inneholder den inverse avstanden. Erfaring fra målfølgning på radar bekrefter at det kan være ugunstig å bruke den inverse avstanden i et system med aktiv sonar. Erfaringene viser at det ikke funnet behov for å gjøre denne konverteringen. Det er derfor rimelig å anta at moderne målfølgingsalgoritmer for aktiv sonar heller ikke har denne konverteringen. Det foreslås derfor å rette dette i målfølgingsalgoritmen.

### 3.2.2 Parametere

Det er ingen parametere som er direkte galt innstilt eller som ikke påvirker systemet slik det er tiltenkt. Parametrene for støy har rom for utbedring. Støymatrisen til prosessen er ikke undersøkt fordi den er dokumentert i (5). Støymatrisen til målingene kan undersøkes nærmere for å se vurdere en mer effektiv bruk av SNR-verdiene.

### 3.2.3 Algoritmen

I funksjonen *hyp\_scoring*, som gjelder vektig av ekko, har Løland innført en *rettelse* av multinormalfordelingstettheten.

$$\mathbf{N}(\mathbf{e}; 0, \mathbf{S}) = |2\pi\mathbf{S}|^{-\frac{1}{2}} \cdot \exp\left\{-\frac{1}{2}(\mathbf{y} - \bar{\mathbf{y}})^T \mathbf{S}^{-1}(\mathbf{y} - \bar{\mathbf{y}})\right\} \quad (3.6)$$

$$\mathbf{N}(\mathbf{e}; 0, \mathbf{S}) \approx (2\pi)^{-\frac{n}{2}} \cdot |\mathbf{S}|^{-\frac{1}{2}} \cdot \exp\left\{-\frac{1}{2}(\mathbf{y} - \bar{\mathbf{y}})^T \mathbf{S}^{-1}(\mathbf{y} - \bar{\mathbf{y}})\right\} \quad (3.7)$$

Det presiseres at (3.7) ikke er en rettelse om gir likt svar. I samtale med Løland ble det klart at (3.7) fører til at man tror mindre på målingen enn den opprinnelige formelen (3.6). Resultatet blir lavere TTP og færre falske track. Det kan spekuleres i om dette ble gjort fordi så mange ekko falt inn i valideringsregionen.

I funksjonen *search\_poss\_upd*, som markerer mulige nye track, så blir det lagt til en korreksjon på hastigheten til det nye, foreløpige tracket når det blir generert av et ekko som er to ping gammelt. Korreksjonen gjelder peilingen:

$$cor = \frac{\alpha_t + \alpha_o}{2} \quad (3.8)$$

Der  $\alpha_t$  er vinkelen mellom startekko som er to ping gammelt, og nytt ekko.  $\alpha_o$  er vinkelen mellom startekko og observatør i inneværende ping. Korreksjonen kommer inn der egenskapene til det foreløpige tracket blir satt:



$$x\_tentative = \begin{bmatrix} ekko_{x,k} \\ ekko_{y,k} \\ d \cdot \cos(cor)/2\Delta T \\ d \cdot \sin(cor)/2\Delta T \end{bmatrix}, d = \sqrt{(ekko_{x,k-2} - ekko_{x,k})^2 + (ekko_{y,k-2} - ekko_{y,k})^2} \quad (3.9)$$

Det er trolig valgt å bruke en korreksjon for å veie opp forflytningen til observatøren mellom pingene. Siden koordinatene til ekkoene er geografisk orientert antas det at en slik korreksjon ikke er nødvendig, men har likevel latt uttrykket stå. Grunnen til det er løsningen som er valgt videre i funksjonen. Når et ekko som er enten ett eller to ping gammelt ligger slik til at det er ett eller flere ekko i innværende ping som kan skape track, dannes en variabelstruktur som inneholder informasjon om det foreløpige tracket. Det er slike ting som track ID, lengde og lignende. Løsningen som er gjort her tillater at kun ett nytt track kan oppstå fra ett ekko. Så hvis det er flere ekko som ligger slik til at de oppfyller betingelsene for dannelsen av et track, og selv om punktene peker i forskjellige retninger så vil gjennomsnittet av alle ekkoene bli brukt som målepunkt for tracket. Løsningen sparer regnekraft siden langt færre track blir opprettet, men skjuler et tracks virkelige egenskaper i oppstarten. En løsning som danner et nytt track for hvert av ekkoene som oppfyller betingelsene, krever minimale endringer i koden. Et tankekors kan være at hvis man skal sjekke alle ekkoene i oppstarten, så hvorfor skal man ikke da sjekke om ekko som inngår i track kan være startpunkt til andre track? Men det bryter med hensikten til IMMPDA som skal låse seg inn på ekte mål, selv om oppstarten kanskje var basert på falske ekko.



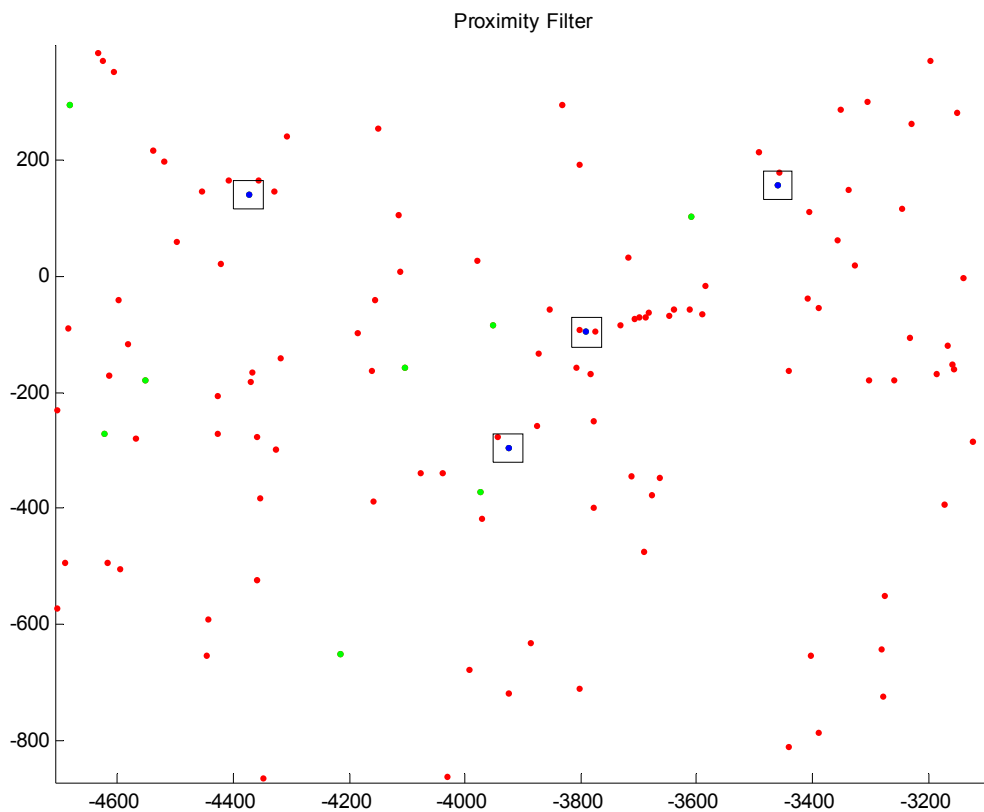
## 4 PROXIMITY FILTER

I kapittel 2.2 er utgangspunktet for et proximity filter nevnt. Dette kapittelet beskriver implementering og testing av filteret.

For å sikre at man fikk et proximity filter som virket slik det var ment å gjøre, ble det valgt å lage et PF som ikke var i forbindelse med målfølgingsalgoritmen. Da filteret hadde den funksjonaliteten som var ønsket, ble den integrert i målfølgeren som en valgfri opsjon.

### 4.1 Frittstående proximity filter

Implementasjonen av filteret var en enkel søkealgoritme som fant ekko som oppfylte betingelsene. Filteret hentet inn de data som skulle filteres, og viste grafisk før/etter-bilde med størrelsen på søkevinduene som vist i figur 4.1.

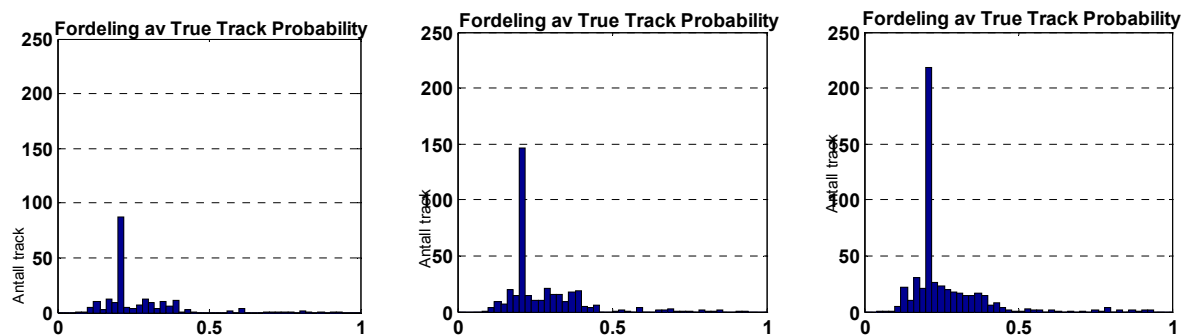


Figur 4.1: Viser resultat av PF

I figuren er blå prikker ekko som passerer filteret, grønne slettes og røde er historiske ekko. Svarte firkanter er søkevinduet med ekko som har passert i midten. Figuren viser resultatet av ett ping kjørt gjennom filteret med 25x25m søkevindu og 10 pings historie.

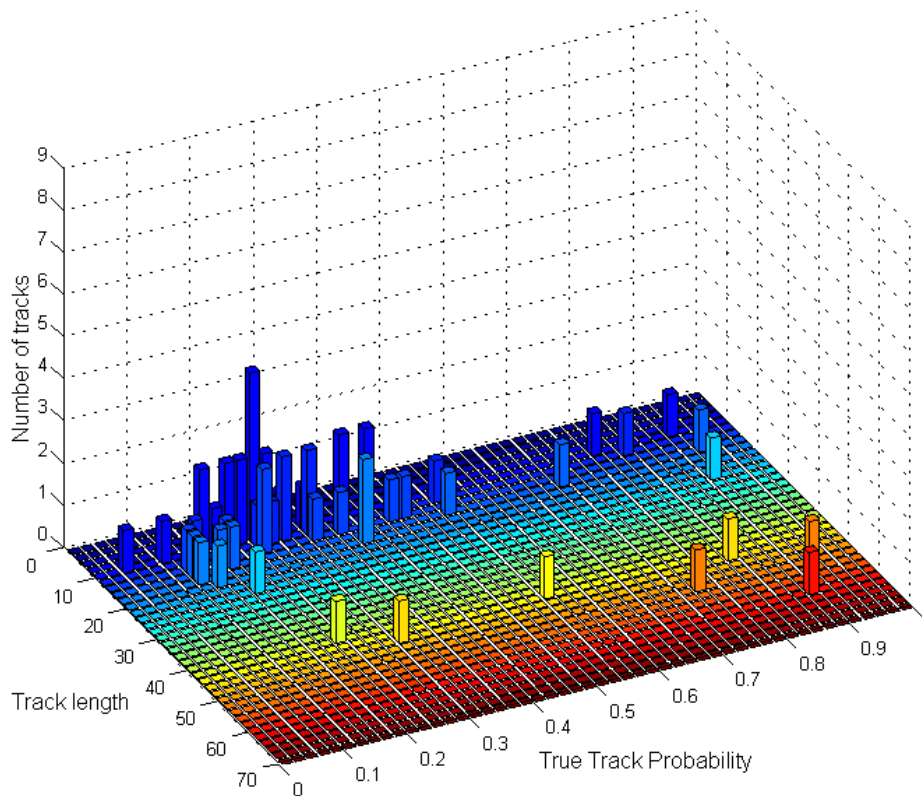
## 4.2 Integrert proximity filter

I den integrerte utgaven ble all grafisk visning utelatt, og mulighet for å endre historiesøket utenfor funksjonen ble lagt til. Endringene i målfølgingsalgoritmen var minimale, kun noen *if*-setninger som sikret at resultatene fra PF ikke ble endret. Det ble testet historiesøk på 2, 5, 7, 10 og 15 ping. Ytelsesmessig er det en kraftig forbedring å kjøre med et PF med historiesøk på 2 ping. Kjøretiden ble halvert. Det har sin grunn i at det blir langt færre ekko å jobbe med. Bakdelen er at målfølgeren danner track på det ekte målet langt senere enn optimalt. Dette skjer fordi track som starter tidligere og som baserer seg på ekko fra det ekte målet, får så lav TTP at de blir slettet.



Figur 4.2: Viser fordeling av TTP ved 2, 5 og 10 pings PF historie

Endringen i fordelingen av TTP viser at lengre historiesøk gir flere track. Likevel er antall track som har TTP over 50% nesten konstant. Historiesøket er derfor en effektiv parameter for å endre antall svake track, dvs TTP < 50%, men det er ingen garanti for at et ekte mål gir et sterkt track. I datasettet brukt her gir det ekte målet et sterkt track, og derfor kan historiesøket stilles lavt. Det fører til at arbeidsbelastningen reduseres, men de interessante trackene forblir. PF bidrar ikke til at track på sterke stasjonære falske mål reduseres. På samme måte som at ekkoene fra det ekte målet ikke blir særlig redusert av filtreringen, blir heller ikke ekkoene, som sannsynligvis kommer på grunn av bunnjenklang, særlig redusert. Dette stemmer med (6).



Figur 4.3: Resultat etter kjøring med historie på 10 ping

En annen illustrasjon av effekten til PF er å vise en figur som viser forholdet mellom TTP, levetid og antall track. En sammenlikning av resultatet før filtrering i figur 3.6 og etter filtrering i figur 4.3, viser at det i hovedsak er track med TTP på under 0.5 og levetid på under 35 som reduseres. Tracket på et ekte målet har blitt noe kortere, men skiller seg fremdeles ut i figur 4.3, som det gjorde i figur 3.6.

Proximity filteret har vist seg å avlaste målfølgeren ved å fjerne ekko som mest sannsynlig har opphav i støy, men denne funksjonen skal finnes i målfølgingsalgoritmen i form av data assosiasjon. Filteret setter en begrensning på øvre tillatt hastighet til mål, men denne funksjonen er også innbygget i målfølgingsalgoritmen. Filteret har en verste kjøretid på data lik  $E(10n^2)$  ved historie på 10, der  $n$  er antall ekko i pinget, så avlastningen på målfølgeren har sin pris. Siden funksjonene til proximity filteret allerede er i målfølgingsalgoritmen, er det derfor ikke være nødvendig å bruke filteret sammen med målfølgingsalgoritmen. For å få PF virkning i målfølgingsalgoritmen an man sette lavere maksimal hastighet, og oppnå tilnærmet samme resultat. PF tilbyr likevel en mulighet for å prosessere data før de anvendes i målfølgingsalgoritmen. Med slike prosesserte data gir målfølgingsalgoritmen raskere resultat som kan analyseres. Det er da viktig at datasettet er kjent apriori, slik at man har kontroll på hvilke data som er filtrert vekk ved hjelp av PF.



## 5 KONKLUSJON

Målfølgingsalgoritmen fra Løland (5), er gjennomgått og sammenliknet med teori for *Interactive Multiple Model Probabilistic Data Association*. Implementasjonen av teorien kan ha forskjellige løsninger som gir tilnærmet samme svar. De fleste av disse løsningene går ut på å slette track for å spare datakraft, men enkelte av disse er endret eller deaktivert under arbeidet på grunn av økt tilgang på regneressurser. Noen endring på målfølgingen er ikke registrert, men man har nå bedre kontroll over hvilke track som blir slettet. Det bedrer igjen muligheten for å ta vare på informasjonen til de slettede trackene.

Standard avvik i avstand som er avstandsavhengig er kjent fra systemer med passiv sonar. Der får man kun peilingen til målet, men det finnes metoder for å finne avstanden. Dette vil gi en avstandsavhengig usikkerhet i avstanden til målet. Det er tvilsom om det er riktig å videreføre dette over til system med aktiv sonar der man får avstanden til målet ut fra kun én måling. Man har ikke grunn til å tro at måleusikkerheten i avstand har særlig skjev normalfordeling. Det er derfor ingen grunn til å bruke den inverse avstanden som tilstandsvariabel. Et resultat av bruken av den inverse avstanden, er at valideringsregionen dekker i verste fall halve datamengden. Det gir et usikkert estimat av målingen som ikke gjenspeiler den usikkerheten som faktisk er i målingen.

Påvirkningen av parametrene er dokumentert. Selv om noen fasit på parameterinnstillinger ikke eksisterer, så gir de innstillingene som er satt god målfølgning for dette datasettet. Testing med flere datasett er nødvendig for å kunne si noe om robustheten til de valgte parametrene.

Proximity filteret ble innført for å filtrere ut ekko som har stor sannsynlighet for å være tilfeldig støy. Det vil si at et geografisk plassert ekko må ha en viss historie av ekko for at et nytt ekko kan passere filteret. Resultatet av målfølgning på data som var filteret, var at kjøretiden på målfølgeren ble merkbart forkortet. Filteret reduserte effektivt track som hadde lav TTP og kort levetid, men blant disse kan det og være et ekte mål. Funksjonaliteten som ligger i filteret finnes også i målfølgingsalgoritmen. Filteret vil være et hjelpemiddel når man skal studere saktegående mål. Ved å filtrere inndataene vil målfølgingsalgoritmen raskere gi de resultatene som skal studeres.

**LITTERATUR**

- (1) Y. Bar-Shalom & T. E. Fortman (1988): Tracking and Data Association In: *Mathematics in science and engineering Vol 179* (Eds W. F. Ames), Academic Press.
- (2) Y. Bar-Shalom (1990): Multitarget-Multisensor Tracking: Advanced Applications, Artech House.
- (3) Y. Bar-Shalom & X. Li (1995): Multitarget-Multisensor Tracking: Principles and Techniques, YBS.
- (4) F. Haugen (1996): Regulering av dynamiske systemer Bind 2., TAPIR Forlag.
- (5) A. Løland (2001): UNISON 2005 - Target tracking in clutter using active sonar, theory and practice, FFI/RAPPORT-2001/00183, Konfidensiell
- (6) J. Wegge & E. Tveit (2001): Correlation between sonar echoes and sea bottom topography. Based on sonar data from KNM Narvik in Bjørnafjorden april 1999, FFI/RAPPORT-2001/04022, Begrenset
- (7) J. Eidsvik & A. Løland (1999): UNISON 2005 - Target tracking in clutter using active sonar, FFI/NOTAT-99/03817, Konfidensiell



## APPENDIKS A

### ○ Endringer på original målfølgingsalgoritme

I *hyp\_merge* er rutinen for å finne det tracket som burde få lov til å fortsette, er basert på at det er det eldste som er det rette. Men det eldste er ikke nødvendigvis det som har den høyeste  $\mu$  (*True Track Probability* (TTP)). Derfor er denne rutinen endret til å sjekke  $\mu$  i tillegg til alderen til tracket. Man begynner da med det nyeste tracket for å få sjekket flest  $\mu$ .

```
% Find the oldest hypothesis, starting with the newest
for j=tsize:-1:1
    if t(j).bool_mat(f(i))>0 % if track use the echo
        j_val(j)=1;
        if (identity > t(j).id)&(tt_prob <= (t(j).my(t(j).length-1)))
            identity=t(j).id;
            tt_prob = t(j).my(t(j).length-1);
            j_best=j;
        end;
    end;
end;%for i=1:size(f,1)
```

I *hyp\_merge* er rutinen for å slette de track som deler et målepunkt og har en differanse i hastighetsretning som er mindre enn den satte terskelen.

```
for l=1:size(w,1)
    if max(diff_course(1:size(w)))<C_T
        del_these(w(l))=1;
    end;
end;
```

Problemet her var at det ble sjekket om den maksimale differansen i  $w$  lå under terskelen, hvis så ble alle slettet. Dessuten ble ikke  $w$  nullstilt mellom analysene, så  $w$  kunne bestå av flere elementer enn det som egentlig var tilfellet. Endringen er vist nedenfor:

```
for l=1:size(w,1)
    if diff_course(l) < C_T,
        del_these(w(l))=1;           % Marking tracks to be deleted
        NOTE=NOTE+1;                % Increases if merging.
    end;%if max...
end;%for l=1:size(w,1)
```

Endringene består i at hvert punkt sjekkes, og at  $w$  nullstilles i starten av *for*-sløyfen, og at NOTE (som inneholder hvor mange track som er blitt slettet) blir oppdatert for hver virkelige sletting.

## APPENDIKS B

### ○ Lagring av målfølgerdata for hvert ping

Som et ekstra hjelpemiddel i analyse av målfølgeren er det lagt inn mulighet for å lagre informasjon om alle track i et ping selv om de er slettet, sammenslått eller kjørende. Dette skjer i en egen funksjon slik at endringer i hvilke data som skal eksporteres kan endres uten større inngrep i koden til målfølgeren.

Filene består av en 16-linjer inndeling der første del er header og kommentarer, mens påfølgende deler er trackinformasjon på formen:

```
Track:                <= Kommentar
Track.id
Ping number
start x               <= Koordinater til første ekko i track
start y
this x               <= Koordinater til siste ekko i track
this y
velocity x           <= Hastighetsvektorer til track
velocity y
True Track Probability
Track length
Status: 0-new track, 1-updated track,2-no update=>using prediction,
3-too high speed=>deleted,4-too low TTP=>deleted,
5-no updates for 10 pings=>deleted,6-merged=>deleted
-   <= Leding plass
-   <= Leding plass
-   <= Leding plass
-   <= Leding plass
```

Filene kan brukes i andre analyseverktøy som studerer andre elementer rundt AU-operasjoner.

## FORDELINGSLISTE

**FFIBM**
**Dato:** 10. mars 2003

RAPPORTTYPE (KRYSS AV)		RAPPORT NR.	REFERANSE	RAPPORTENS DATO	
<input checked="" type="checkbox"/> RAPP	<input type="checkbox"/> NOTAT	<input type="checkbox"/> RR	2003/01001	FFIBM/849/166	17. mars 2003
RAPPORTENS BESKYTTELSESGRAD			ANTALL EKS UTSTEDT	ANTALL SIDER	
UGRADERT			32	41	
RAPPORTENS TITTEL			FORFATTER(E)		
INTERACTING MULTIPLE MODEL PROBABILISTIC DATA ASSOCIATION – En studie av målfølgingsalgoritme for aktiv sonar			GJERSØE, Amund Lorentz		
FORDELING GODKJENT AV FORSKNINGSSJEF			FORDELING GODKJENT AV AVDELINGSSJEF:		
Elling Tveit			Jan Ivar Botnan		

**EKSTERN FORDELING**
**INTERN FORDELING**

ANTALL	EKS NR	TIL	ANTALL	EKS NR	TIL
1		KNM Tordenskjold	9		FFI-Bibl
1		KL Odd Erling Furu	1		FFI-ledelse
1		KL Odd Sverre Oma	1		FFIE
1		KL Atle Knudsen	1		FFISYS
1		Førstekonsulent Wendy Hartog	1		FFIBM
			1		FFIN
1		FLO/Sjø	1		Forfattereskemplar(er)
1		KL Harald Tholo	1		Restopplag til FFI-Bibl
1		Senior ing. Svein Mjøsnes	2		Avd kontor FFIBM/Horten
					<b>Elektronisk fordeling:</b>
1		KE			FFI-veven
1		KK Halvard Flesland			Stig Lødøen (SEL)
1		KK Hans Petter Midtun			Arvid Melkevik, (AMk)
1		OK Jan Ove Skorpen			Tor Knudsen (TKn)
					John-Mikal Størdal (JMS)
		Kongsberg Simrad			Erik Lofstad Ahlsen (ELA)
1		John Glattetre			Svein Alsterberg (SAI)
		Postboks 111			Thomas Martin Dørum (TMD)
		3191 HORTEN			Brita H Hafskjold (BHa)
					Frederik Hermansen (FrH)
		KDA			Karl T. Hjelmervik (KTH)
1		Espen Mathisen			Einar Høst (EiH)
		Postboks 1003			Erik Nordø (ERn)
		3601 KONGSBERG			Elling Tveit (ETv)
					Jon Wegge (JWe)

FFI-K1

Retningslinjer for fordeling og forsendelse er gitt i Oraklet, Bind I, Bestemmelser om publikasjoner for Forsvarets forskningsinstitutt, pkt 2 og 5. Benytt ny side om nødvendig.